MAGELLAN RECORDER DATA RECOVERY ALGORITHMS

Chuck Scott
Project Test Section
Jet Propulsion Laboratory
Pasadena, Ca 91109

Howard Nussbaum
Radar Systems
Hughes Aircraft
Los Angeles, Ca 90009

Scott Shaffer
Radar Science Section
Jet Propulsion Laboratory
Pasadena, Ca 91109

ABSTRACT

This paper describes algorithms implemented by the Magellan High Rate Processor to recover radar data corrupted by the failure of an onboard tape recorder that dropped bits. For data with error correction coding, an algorithm was developed that decodes data in the presence of bit errors and missing bits. For the SAR data, the algorithm takes advantage of properties in SAR data to locate corrupted bits and reduce there effects on downstream processing. The algorithms rely on communication approaches, including an efficient tree search and the Viterbi algorithm to maintain the required throughput rate.

KEYWORDS

Error Correction, Frame Synchronization, Viterbi Algorithm

INTRODUCTION

When NASA's Magellan spacecraft was launched on May 4,1989, it's primary mission was to map the surface of Venus using Synthetic Aperture Radar (SAR). Magellan arrived at Venus August 10, 1990, and began it's primary mission. In three years of operation, Magellan has successfully imaged over 98% of the planet's surface, and collected more data than all previous planetary missions combined [Ref. 1]. To date the spacecraft has returned over 700 gigabytes of data to the Jet Propulsion laboratory (JPL) which manages the mission for NASA. However, approximately 50 gigabytes of this data was corrupted by the failure of an onboard tape recorder which dropped bits.

This paper describes the algorithms implemented in the. Magellan High Rate (MHR) processor to recover this data, The next section describes the MHR function and is followed by the description of the tape recorder failure, This is followed by sections that describe the recovery algorithms. These algorithms consist of a modified frame synchronization algorithm, a SAR burst (SAB) header decoding algorithm, and a SAR data recovery algorithm. The SAB header decoding and SAR data recovery use communication approaches to detect the positions of missing bits. The last section discusses the success of the. algorithms at recovering of the corrupted data.

MHR PROCESSOR FUNCTION

Because Magellan's high gain antenna is shared by both the radar and telecommunications subsystems, the SAR data can not be transmitted to Earth directly. Data are recorded by two spacecraft Data Management System (DMS) Recorders, each containing four tracks, A nominal mapping orbit only requires four tracks to record data, but both recorders were used to prevent data loss at track changes. After each mapping pass is complete, data are played back to one of three Deep Space Network (DSN) tracking stations. This telemetry is divided into frames called Radar Composite Data (RCD) frames and are placed on a Original Data Record (ODR) tape.. The MIDR operates on the telemetry to produce data that is needed to form images,

The MIDR consists of 4 functions as shown in Figure 1. The first step in processing the RCD frame is to use frame synchronization to locate a code at the beginning of each RCD frame. "Then, the overlap removal eliminates redundant data from spacecraft recorders and different DSN tracking stations to produce unique frames, This is followed by a second synchronization step to detect SAR burst (SAB) frames. SAB frames are not of a uniform size and change dynamically during the course of an orbit and, therefore, require detection of a SAB synchronization code. Once a SAB frame is found, a SAB header, that contains important information for both creating the SAB frame and the SAR image, is corrected of transmission errors through parity check coding. The SAD frame is then extracted and recorded on an Experimental Data Record (EDR) tape which typically consists of 250 Megabytes of data per orbit.

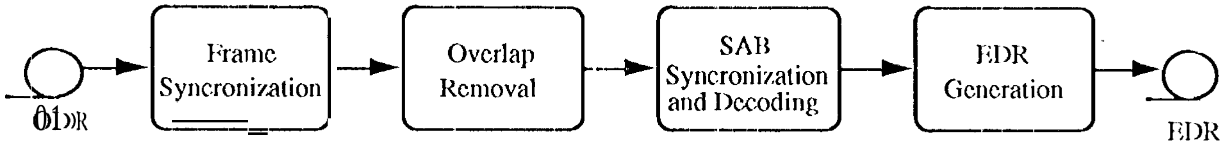

Figure 1 - MIDR Processing Flow

TAPE RECORDER FAILURE

The first indication of a problem with the onboard tape recorder, DMS A, appeared during the cruise from Earth to Venus. When low rate engineering data recorded on DMS A, track 2 was played back to Earth, a high number of frames were discarded by the ground system. All other recorder tracks appeared nominal. 1 examination of the track 2 data showed that the 32 bit frame synchronization code at the begin of discarded frames were corrupted, The corruption caused the bit pattern 1001 to change to 111. This phenomena was dubbed the flip-a-bit-slip-a-bit problem because one bit flips from a zero to a one, and the following bit is dropped, causing the remainder of the frame to slip to the left by one bit,

Two months into the mapping mission, the problem appeared in the high rate radar data recorded on DMS A track 2. Not only did the synchronization codes between RCD frames contain the flip-a-bit slip-s-bit corrupt ion, the shortened frame length bet ween synchronization codes implied that bits within the radar data fields were also corrupted. A comparison of data recorded simultaneously on both DMS A and B confirmed that DMS A was corrupting the radar data in a similar manner. When first detected in the high rate data, the frequency of missing bits was approximately once per 50 Mbytes of data, and had little effect on the resulting radar images, Unfortunately, the error rate steadily increased and gradually spread to the other 3 tracks on DMS A, By the time DMS A was turned off, the rate had increased to over 50 dropped bits per 840 bytes.

FRAME SYNCHRONIZATION

The MIDR frame synchronization software uses a 32 bit synchronization code at the beginning of each RCD frame to locate frame boundaries, The software detects sync codes with up to 7 bit errors and skips (or fly wheels) up to 3 frames without sync codes to maintain lock on successive frames. When frame lock is lost, the software returns to the last good synchronization code and searches bit by bit until the next synchronization code is detected. in order to handle corrupted data, the software was modified (o detect sync codes that are missing a bit. This check is done in parallel with the normal synchronization code check. The software was also modified to begin searching for synchronization codes **15 bits** before the expected location. If more than one synchronization code is detected in the following 15 bit region, the position producing the fewest bit errors is selected. If a synchronization code is not detected, the software continues as before. **1** Detected frames are passed on to the remove overlap module along with their frame length.

To properly interpret the SAR data, SAB headers must be located by detecting the synchronization code at the beginning of each header. Locating the SAB is similar to the frame synchronization process except that the SAB varies in length. The number of positions which are searched to detect the SAB sync code is limited by the number of missing bits in the RCD frame containing the header. The SAB synchronization algorithm is enhanced by the ability to verify the SAB header detection by successfully decoding the total SAB header using the algorithm described in the next section,

SAB HEADER DECODING

The SAB header contains information necessary to interpret the SAR data during processing of a SAR coherent array. If the SAB header is corrupted, the associated data is not processed, and the SAR image strip will have a gap. Recognizing the importance of the SAB header, error correction parity check bits are included to protect the information from bit errors . Fortunately, the parity check coding is also useful to counter missing bits in the SAB header. 'l'his section describes an algorithm [Ref. **2]** to decode the SAB header in the presence of missing bits that was implemented by Magellan system.

The SAB header consists of **54** bytes of information bits and **54** bytes of parity check bits. 'The encoding is achieved using a Golay (24,12) code that has the capability to detect and correct up to 3 bit errors and detect up to 4 errors. 24 interleaving is used to order the sequence of bits $\{x_n\}$ in the header into 36 codewords of length 24 bits. Figure 2 shows the interleaving scheme described by the matrix of 36 columns and 24 rows. Each column of the matrix is a codeword where syndrome decoding is used to correct errors. A missing bit in the sequence causes a skewing of the data where a column of data will no longer contain data from a single codeword,

$$\begin{array}{c} \text{36 Columns} \\ \left[\begin{array}{ccccc} x_1 & x_2 & \cdot & \cdot & x_{36} \\ x_{37} & x_{38} & \cdot & \cdot & x_{72} \\ \text{Rows} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{829} & x_{830} & \cdot & \cdot & x_{864} \end{array}\right] \end{array}$$

Figure 2- Data Matrix

Syndrome decoding, which is the basis for decoding an individual codeword, first forms a syndrome vector, $\underline{s}$, by

$$\underline{s} = \underline{H}\underline{x}\Big|_{\text{mod } 2}$$

where $\underline{H}$ is the parity check matrix of dimension 24 columns by 12 rows and $\underline{x}$ is a codeword, Then, a table lookup determines an error vector that is used to correct the data,

The basic algorithm for decoding the SAB header in the presence of missing bits is to form hypotheses for all possible patterns of missing bit positions where a pattern has up to M missing bits. ']'hen, decoding is performed for each hypothesis. This decoding is achieved by inserting bits in the missing bit positions to adjust the data prior to decoding. The synchronization code at the beginning of the SAB header guarantees that the beginning of the data is correctly aligned and that the insertion of the missing bits only affects the alignment of the bits after the missing bits. The accepted decoded data corresponds to the set of missing bit positions that have. the lowest total bit errors for the 36 codewords.

This approach requires an excessive amount of computation, 864 SAB decodings for a single missing bit and 374,113 for two missing bits. This excessive amount of computation was greatly reduced by developing an algorithm that reduces the number of syndrome vectors computed. The algorithm relics on three features:

(1) Grouping the individual hypotheses to allow pruning the hypotheses
(2) Using a tree search to eliminate the necessity of evaluating fully developed hypothesis
(3) Evaluating the syndrome vectors efficiently

With this algorithm, decoding in the presence of up to 4 missing bits was successfully implemented for the Magellan application.

The first feature of the algorithm is to group hypotheses of missing bit positions into subsets of hypotheses.' A subset of the hypotheses is a collection of hypotheses where the missing bit positions occur on specific rows of the data matrix of Figure 2. For example, consider the subset of hypotheses where each hypothesis has a missing bit occurring in row i and row j. The totality of the subsets is all row combinations where missing bits can occur.

By using this subset grouping, onc first finds the best missing bit position with the lowest number of detected bit errors for each subset, This is achieved by employing a dynamic programming approach described below. Finally, with the knowledge of the number of errors associated with all possible row combinations where missing bits can occur, the algorithm selects the row combination with the, lowest number of errors. Thus, the missing bit pattern for the selected row combination is the accepted pattern, and the associated decoded data is the decoded SAB header.

The approach of evaluating a subset using dynamic programming is efficient because each of the hypotheses within a subset of the hypotheses dots not have to be fully evaluated. To illustrate this assertion, consider a situation where one bit is missing and the hypothesis is that the bit is missing in row i. Now evaluating the number of errors for the codeword in column 1, either the missing bit occurred in column 1 or after column 1. Onc constructs the two codewords for these two statements by appropriately adjusting the codewords associated with column 1, and evaluates the syndrome vector and determines the number of errors for the two codewords. The next step is to apply the following recursion for the other columns to determine the cumulative error count at each column where the cumulative error count for the nth column is the sum of the detected errors for the first n columns. Suppose for the n^{th} column, one knows the missing bit position occurring in one of the first n columns in row i that has the smallest cumulative error count, and the cumulative error count assuming the missing bit occurs after the n^{th} column in row i. Combining this information with the number of errors associated with the two codewords in the n+ 1st column that represent a missing bit occurring in row i before column n+ 1 and represent a missing bit after column n-I 1, one computes the missing bit position occurring in the first $n+1^{st}$ columns that has the lowest cumulative error count, Also, the cumulative error count assuming the missing bit occurs after the n-i 1st column in row i is computed. When the update for column 36 is computed, the best missing bit position is determined for a missing bit occurring in row i . This procedure eliminates hypotheses that have a higher error count than the missing bit position of the final update. In this way, onc has to evaluate the syndrome vector only twice per column as opposed to 36 t imes per column for the brute force approach.

The recursion in the algorithm is best described by a state transition diagram where the number of syndrome vectors evaluated pcr column to update the states is 2^M where M is the number of missing bits. For example assuming there are 2 missing bits, the state transition diagram is shown in Figure 3. At each node, a decision is made as to which branch has the lowest cumulative error and the associated missing bit pattern is assigned to the state. The recursion described by the state diagram must be evaluated 35 times, once per column for each subset (row combination).

Further efficiency in the evaluation is achieved by using tree search procedures, the second algorithm feature. Instead of evaluating each row combination for the full 36 columns, one is more judicious. The recursion to extend the column for a row combination is only applied to the row combination with the lowest number of cumulative errors until one of the row combinations is fully evaluated and the tree search terminates. Thus, each of the row combinations will be evaluated to a different column depth. This procedure was found to be efficient, since the bit error rate was low and most of the row combinations are not extensively evaluated. Figure 4 shows a block diagram of the processing where the evaluation of the error count is discussed next.

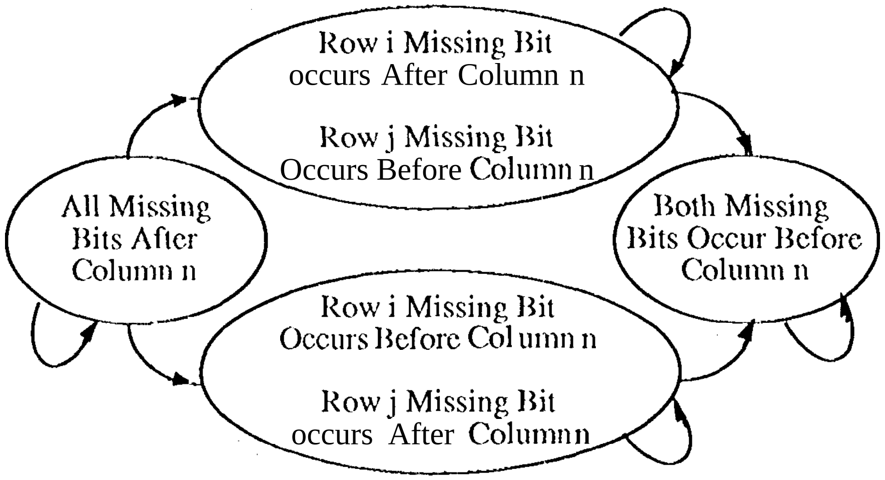
Row i Missing Bit occurs After Column n
Row j Missing Bit Occurs Before Column n
All Missing Bits After Column n
Both Missing Bits Occur Before Column n
Row i Missing Bit Occurs Before Column n
Row j Missing Bit occurs After Column n

Figure 3 - State Transition Diagram For 2 Missing Bits

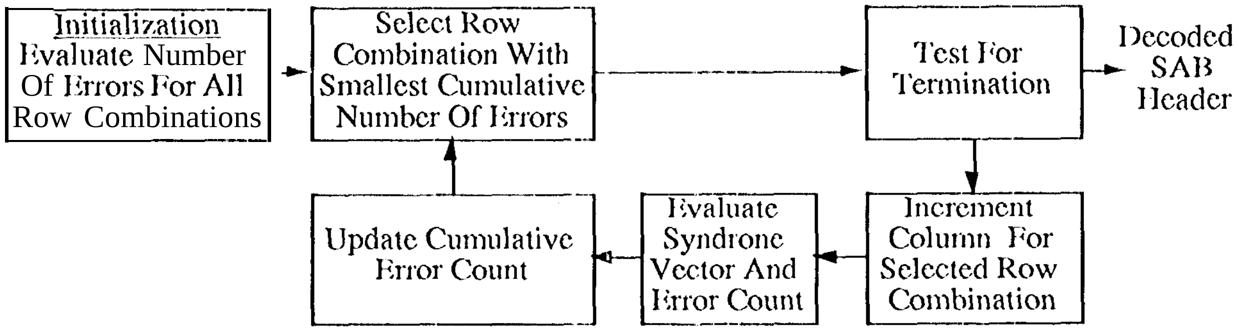
Initialization
Evaluate Number Of Errors For All Row Combinations
Select Row Combination With Smallest Cumulative Number Of Errors
Test For Termination
Decoded SAB Header
Update Cumulative Error Count
Evaluate Syndrone Vector And Error Count
Increment Column For Selected Row Combination

Figure 4- Block Diagram Of Algorithm

In order for the tree search to be efficient, the evaluation of the syndrome vector must also be efficient, the third algorithm feature. This is achieved by evaluating at the" beginning of the algorithm quantities that would normally be evaluated many times in the algorithm to evaluate syndrome vectors. These quantities are

$$\underline{\Delta}_n(i) = \underline{h}_i\{x_{36i+n} \oplus x_{36i+n-1}\}\big|_{\mathrm{mod}\,2} \qquad i = 1,...24;\ n = 1,...36$$

$$\underline{\Sigma}_n(i) = \sum_{j=i}^{24} \underline{h}_i x_{36j+n}\Big|_{\mathrm{mod}\,2} \qquad i = 1,...24;\ n = 1,.,.36$$

where $\underline{h}_i$ is the ith column of the parity check matrix $\underline{H}$.

To illustrate the use of these quantities, consider the case of 2 missing bits and define the syndrome vector $\underline{s}_n(i,j)$ to be the syndrome vector for the nth column when bits are inserted in the data sequence in rows i and j before column n to correct for missing bits. For initialization of the tree search, syndrome vectors for the first column and all row combination of missing bits are evaluated using the recursion

$$\underline{s}_1(25,25) = \underline{\Sigma}_1(1)$$

$$\underline{s}_1(i,25) = \underline{s}_1(i+1,25) \oplus \underline{\Delta}_1(i) \qquad i = 1,...24$$

$$\underline{s}_1(i,j) = \underline{s}_1(i,j+1) \oplus \underline{\Delta}_{36}(j-1) \qquad i = 1,...24;\ j = i,...24$$

where a missing bit occurring in row 25 signifies that no missing bit occurred. Besides the initialization, the update of the state transition diagram requires the evaluation of a series of syndrome vectors. For the case of 2 missing bits, the syndrome vectors needed to update the states for the nth column and for the subset corresponding to missing bits in rows i and j are given by

$$\underline{s}_n(i,j) = \underline{\Sigma}_n(1) \oplus \underline{\Sigma}_n(i) \oplus \underline{\Sigma}_{n-1}(i) \oplus \underline{\Sigma}_{n-1}(j) \oplus \underline{\Sigma}_{n-2}(j)$$

$$\underline{s}_n(i+1,j) = \underline{s}_n(i,j) \oplus \underline{\Delta}_n(i)$$

$$\underline{s}_n(i+1,j+1) = \underline{s}_n(i+1,j) \oplus \underline{\Delta}_{n-1}(j)$$

$$\underline{s}_n(i,j+1) = \underline{s}_n(i+1,j+1) \oplus \underline{\Delta}_n(i)$$

where the ordering of the equations is in the form of a Gray code.

SAR DATA

A missing bit in an RCD frame containing SAR data could corrupt nearly the entire frame, since bits after the missing bits will be misinterpreted. Misinterpretation of this amount of data would highly degrade the SAR imagery. To prevent this corruption, positions of missing bits are estimated and bits are inserted to correct the integrity of the. bit alignment. Fortunately, the image fidelity is tolerant of inaccuracy in estimating the position of the missing errors. This reduces the requirements on the algorithm to detect missing bits and allows a statistic communication approach to be applied. The statistical communication approach models the data with a communication channel and applies the Viterbi Algorithm for convolutional codes to estimate the positions of the missing bits. A more elaborate approach that examines the effect of missing bits on the imagery was rejected because of complexity.

The communication channel that models the missing bits relies on the characteristics of the radar. The radar samples the radar return in inphase (I) and quadrature (Q) form using two 8-bit analog-to-digital converters. Then, the Block Adaptive Quantizer (BAQ) quantizes each I and Q 8-bit sample to two bits, which are a sign bit and a magnitude bit, Hughes Aircraft developed and analyzed the performance of the BAQ that is chronicled in Ref. 3. The effect of the BAQ is to make the statistics of the sign and magnitude bits regular and independent of the scene content, where the sign bit is 0 with probability P_s (.5) and the amplitude bit is 0 with probability P_m (.67). Furthermore, the characteristics of the SAR data, before processing, contribute to making the statistics of the samples stable and independent.

The algorithm to detect the missing bits relies on the difference in probability between the sign and magnitude bits. This difference is sensed by noting that the bit pattern in a byte where an even number of bits has been dropped prior to the byte will alternate sign then magnitude bits, while if an odd number of bits is dropped prior to the byte, the bit pattern alternates magnitude then sign bits.

The basics algorithm defines a sequence associated with the data stream that describes whether an even number of bits is missing prior to a corresponding pair of bits. An information bit is associated with each pair of data bits in the data stream where the information bit is O if an even number of bits is missing prior to the associated pair of data bits, and the information bit is 1 when an odd number of bits is missing, With this definition, a discrete memoryless communication channel, depicted in Figure 5, describes the model. I'hen, the Viterbi Algorithm is used to determine the maximum likelihood sequence where it will be shown that the sequence determination can be described by a trellis diagram, Finally, the positions of the missing bits occur at the end of the runs, transition from O to 1 or 1 to O, of the maximum likelihood sequence of information bits.

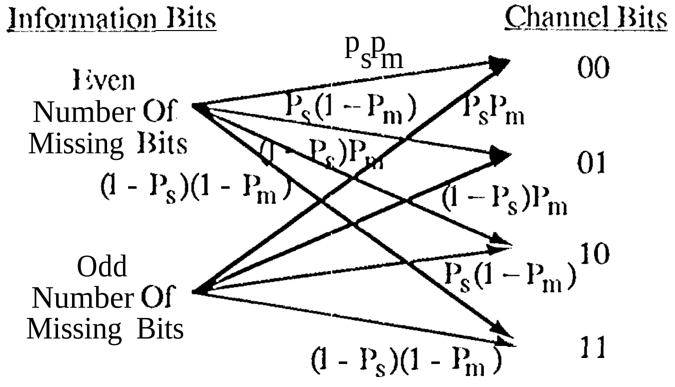

Figure 5- State Transition Diagram

The production algorithm only determines the position of the missing bit to within a byte boundary. This simplifies the processing without degrading the performance on the final SAR image since the accuracy of the estimate position is 5 bytes. With this restriction, define the sequence (i(n)) to be the information symbols where i(n) corresponds to the number of missing bits prior to the nth byte, b(n), where there are N bytes in the sequence. The conditional probability, that the likelihood function is based on, is given by

$$\Pr\{b(n)|i(n)\} = \left(\frac{P_m(1-P_s)}{P_s(1-P_m)}\right)^{\mathrm{mod}_2(i(n))A(b(n))} \Pr\{b(n)|i(n)=0\}$$

where A (b(n)) is the difference bet ween the number of 1s in even bits of b(n) and the number of 1s in odd bits of b(n). The equation is in a form to show that only the first term depends on the information symbols i(n).

The log-likelihood function, which is used to find the most likely sequence {i*(n)), is given by

$$\ell(\{i(n)\}) = \sum_{n=1}^{N} \ell n\{\Pr\{b(n)|i(n)\}\}$$

Upon substituting the conditional probability and eliminating terms independent of i(n), the sufficient statistic of the log-likelihood function is simply given by

$$L(\{i(n)\}) = \sum_{n=1}^{N} \mathrm{mod}_2(i(n))A(b(n)).$$

'1'hen, the problem reduces to finding the sequence {i*(n)) which has R runs and maximizes L({i(n)})) where R is the number of missing bits in an RCI) frame plus 1 The positions of the missing bits are at the byte boundaries corresponding to the ends of the runs in the sequence {i^*(n)).

The determination of the optimum sequence {ix:(n)) is easily found because the metric L({ i(n))), an additive metric, is in a form that the Viterbi algorithm for convolution codes is applicable. The trellis diagram for the Viterbi algorithm for this problem is shown in Figure 6. Each branch is labeled with an information bit and a distance. At each node, a sequence of information symbols and a metric associated with the sequence is determined for each branch going into the node. The node then selects the sequence (branch) with the largest metric. The sequence associated with a branch is determined by extending the sequence associated with the node where the branch emanates from with the information symbol attached to the branch. The corresponding metric of the branch is computed by summing the metric associated with the node where the branch emanates from and the distance attached to the branch. Using this update procedure for the information sequence and metric at each node, the maximum likelihood sequence is found when the trellis is fully evaluate.d.

The implementation of the algorithm proved to be very efficient and effective at estimating the position of the missing bits. The efficiency of the algorithm is based on the property that the algorithm complexity grows only linearly with the number of missing bits.

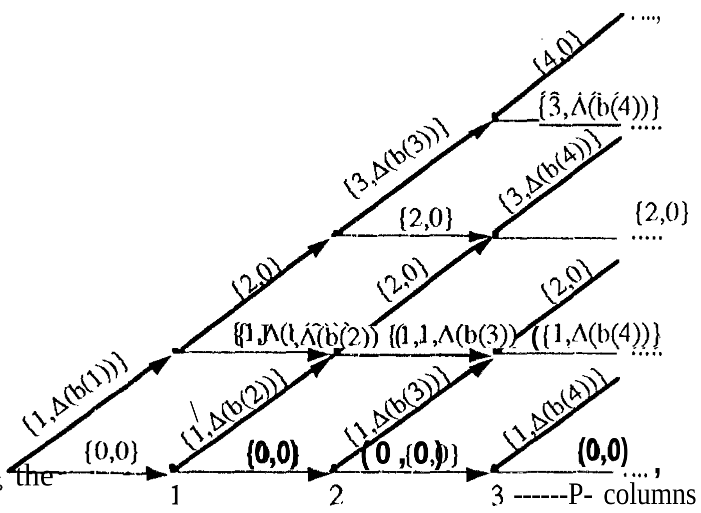

Figure 6 - Trellis Diagram

RESULTS

The algorithms described above successfully recovered data from 378 of the 415 corrupted orbits. The additional time required to decode the SAB header and correct the SAR data was easily offset by the software's ability to maintain frame and SAB synchronization across the corrupted data, only the most corrupted data required longer than nominal to process.

CONCLUSION

An important component of the Magellan mission to Venus is the collection of a global data set. Re-imaging the gaps generated by the tape recorder failure would have required costly mission re-planning and would have competed with other scientific goals. The algorithms described in this paper provided an effective way to fill in the image gaps and greatly reduced the area which had to be re-mapped,

ACKNOWLEDGMENTS

The work described was performed under NASA contract number NAS7-918 at the Jet Propulsion Laboratory, and contract number 959055 at Hughes Aircraft Corporation.

REFERENCES

1. Saunders, R. S., et al., "Magellan Mission Summary," journal of Geophysical Rcasearch, Vol.97e8, Aug 25, 1992, pp. 13067-13090.

2., Nussbaum, H., "SAB Decoding In "The Presence Of Slip-a-Bit Errors," HS513A-0014&0015, Hughes Aircraft, April 1992.

3. Kwok, R. and Johnson, W. T. K., "Block Adaptive Quantization Of Magellan SAR Data," IEEE Trans.Geo.and.Remote Sens., GE-27, July 1989, pp. 375-383.